Deep Adaptation Networks for Summarization

Irene Li[1],Dragomir Radev[1]

[1]Department of Computer Science, Yale University, New Haven, CT.

LILY Lab

Introduction

Summarization has been a very attractive research domain in natural language processing for many years. However, not too many efforts have been done in a domain adaptation scenario. In this project, we try to study and propose a model on how to learn shared features from both source and target domains, and then improve performance on target domain when there are no or fewer summarizations in the target domain.

Domain adaptation is a particular case in transfer learning. Figure 1 shows a use case in domain adaptation. A typical setting would be we have no summarizations in the target domain, and by applying domain adaptation techniques, we are about to improve the performance. The challenges would be, the two domains have frequency biases: the same word appears differently; context feature bias: the same word in different domains means different things. General methods in domain adaptation including instance re-weighting: to focus more on "similar" cases from both domains and learning shared features and attributes.

Materials and Methods

In our project, we will use New York Times Annotated Corpus (NYT-annotated)) dataset for experiments. For each sample, there is the whole article, category (News, Opinion, etc) and its summarization written by human experts. We consider "News" and "Opinion" as two different domains.

The state-of-the-art work on abstractive summarization is the Pointer-Generator Network, which is an attentional sequence-to-sequence model to generate tokens sequence based on a trade-off of generating new words and enhance the original article. An overview shows in Figure 2, where the shared feature layers we are using is the pointer generator network. We will apply the Maximum Mean Discrepancy (MMD) component to the existing summarization network, aiming to minimize the discrepancy of the learned features from the two domains. A discrepancy metric between the source and target is required to bound the target error by the source error, which explored as the two-sample problem.

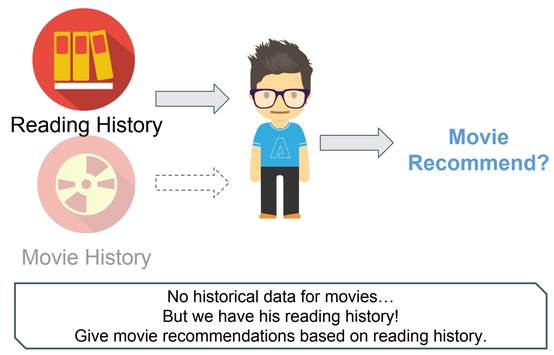

Figure 1. A use case in domain adaptation: Cross-domain Recommendation

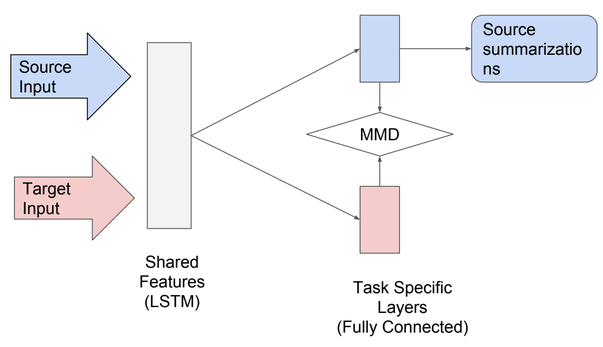

Figure 2. Model overview.

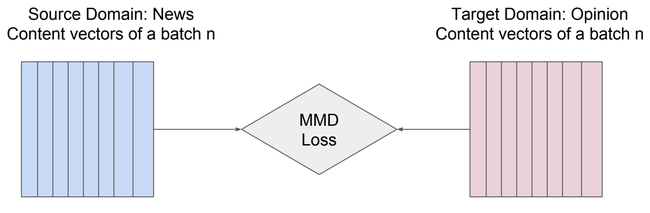

Figure 3. A detailed overview of MMD.

Two-sample Problem

$$X := \{x_1, x_2, .., x_m\} \sim p \text{ and } Y := \{y_1, y_2, .., y_n\} \sim q,$$
$$\text{test whether } p = q$$

MMD Loss

$$\mathrm{MMD}^2[\mathcal{F}, p, q] = \mathbf{E}_{x,x'}\left[k(x,x')\right] - 2\mathbf{E}_{x,y}\left[k(x,y)\right] + \mathbf{E}_{y,y'}\left[k(y,y')\right]$$

A simplified estimation

$$MMD_u^2[\mathcal{F}, X, Y] = \frac{1}{m(m-1)}\sum_{i \neq j}^{m} h(z_i, z_j)$$
$$h(z_i, z_j) := k(x_i, x_j) + k(y_i, y_j) - k(x_i, y_j) - k(x_j, y_i)$$

Gaussian Kernel (RBF), bandwidth could be estimated.

$$K(x, y) = exp(-\frac{||x - y||^2}{2\sigma^2})$$

Loss Function

$$\min_{\Theta} \frac{1}{n_s} \sum_{i=1}^{n_s} J(\Theta(x_i^s), y_i^s) + \lambda MMD^2(D_s, D_t)$$

Pointer generator Loss Supervised	MMD Loss Semi-supervised

Figure 4. The final loss function: supervised part and semi-supervised part.

Discussions about MMD

Figure 3 shows a detailed overview of how to apply MMD in our network model. We take some samples of both domains and feed them into the learned LSTM and get the representations of each. Then we treat them as samples of both domains and apply the MMD to measure how similar they are.

We provide other mathematical details on how to calculate MMD loss. Forming the two-sample problem, we have the original definition. We first project the representations into a feature space using RBF kernel function. We provide a simplified estimation of it which could reduce the computational complexity from n square to n. By minimizing the MMD loss, the parameters in the network will be updated.

As MMD is to measure "distances", the range of the value goes from 0 to infinity. Although in the loss function, the two terms have the same gradient descending direction, the ranges vary during training. So it is crucial to define the weight of each term in the loss function. We come up with a dynamic method: at the beginning, the weight for MMD is small, after some iterations, when the shared feature layers tend to be stable we will increase the weight for MMD, targeting to let the model focus more on target domain samples.

Future works

There are extremely limited works on summarization in transfer learning or domain adaptation. During the whole process, we are facing many challenges. Up to now, we are trying different ways to measure the domain discrepancy. Although MMD showed promising results in computer vision field, and especially for classification tasks. Some efforts still need to be done if it will have good performance in generative models in natural language processing tasks.

Acknowledgement

Special thanks to Prof. Drago, who suggested very helpful points during the whole research. I am also grateful to my friends who helped me with math problems.